ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

Strengthening the Proof of Retrievability with Secure Public Auditing in Cloud Computing

[1] Darubhaigari Ali Ahammed [2] A. Ananda Rao [3] P. Radhika Raju [4] G. Ramesh
[1] M.Tech student [2] Professor [3] [4] Lecturer
[1][2][3][4]Dept of CSE, Jawaharlal Nehru Technological University, Anantapur

***Abstract:* -- Cloud computing provides resources sharing and to handling applications on internet without having local or personal devices. This paper strength the Proof of Retrievability model (PoR) of dynamic data integrity verification on distrusted and outsourced storages on cloud computing. The Outsourced Proof of Retrievability (OPoR) system focuses on cloud storage server for prevention of retune attacks and malicious operations of servers. In Public verifiability the security monitoring is taken by cloud audit server for reducing over head on clients. There is also need to strengthen the secure process of cloud audit server (CAS). This can be provided by generating unique temporary key for each update or modification of file from user. The reset attacks of CAS and cloud storage server (CSS) secure by unique temporary key and deleting the local host replica after verifying the uploaded proof tags of CAS and CSS. And reduce the cost of memory and process time using Elliptic curve cryptography The proposed system strengthening the proof of retrievability (SPoR) model will toughen the resistant of retrievability on upload and update of file operations on cloud computing.**

***Index Terms*— cloud audit server, cloud computing, cloud storage server, integrity, proof of retrievability**

I. INTRODUCTION

The software applications and database moves towards cloud computing which have centralized huge data centers. Because of its advantages like as demand on service, cheap cost of service, high performance, availability, scalability etc. This innovative paradigm brings a many challenges. One of the challenges is data veracity verification at semi trusted services. The servers can delete the file of data which was not access by the users for a long time. In this way servers can delete the expired user's accounts and data for reducing data storage cost. This issue brings distrust worthy on services for storing data in outsource. The service gives the related data to user on their request without having original data. For to solving this problem the users can monitor the data integrity continuously. This process was overloading the work on client for maintenance of data integrity.

In cloud computing platform the data is stored on outside resources then how can the client verify integrity of file without having local replica of data files. There many schemes are developed for Proof of Retrievability (PoR) model [1], [2], [3], [4], [5] to maintain the data integrity on dynamic data operations. The verification procedure is categorized into two ways that are public verifiability and private verifiability. Under public Verifiability the verification progression is undertaken by third party and reduce the over load process on client side. In the private verifiability undertaken by its clients, here they have computational burden for monitoring integrity of data. Another foremost concern is dynamic data operations on cloud data like as modification, insertion and deletion operations on file data. There many models developed for supporting dynamic operations of data on cloud but till now that provide limited services. This paper presents a skeleton of public verifiability with dynamic operations on cloud data storage.

II. RELATED WORK

The provable data possession was first model to propose proof-of-storage scheme based on RSA homomorphic tags for outsource data auditing and support public verifiability. The extension from static data storage to dynamic storage data brings many security problems. In [6] proposed challenge-respond-protocol under the scenario of distributed dynamic storage. It can locate the possible errors and data correctness. In [7] introduce deduplication cloud storage for to save storage space. In PoR model adopted retrievability and possession of information files, but it not support public verifiability scheme. Dynamic PDP model first time explored in [8] using rank-based authentication skip list but it losing desirable efficiency and the computational process overload on clients. The OPoR model can resist the reset attacks which can perform by cloud storage server. It support the public verifiability with dynamic operations using MHT tree, but it constructed under the assumption of trustworthy

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

third party and un trusted cloud storage server. This paper can strengthen the OPoR model construction by under assumption of both miscellaneous parties and resist reset attacks by them. Elliptic curve cryptography [9], [10] proposed a high level security with less bit length of key pair generation.

III. OPOR SYSTEM MODEL

The basic cloud storage data architecture has three basic network entities that are client, cloud storage server, cloud audit server. Client is an entity, which store their large data on cloud and retrieve required data from server for use. It can be either individual organization or consumer. Cloud Audit Server is a third party, which can reduce the computational over head of clients and work as mediator between client and cloud server.

Cloud Storage Server is an entity, which can have storage space and significant resources for database operations. It is managed by cloud service provide. The OPoR model architecture considered with two servers CAS and CSS. It can support public verifiability with dynamic operations in cloud stored data. It can avoid the reset attacks of miscellaneous server, but there is chance that CAS can perform miscellaneous activities and besides this paper can avoid the retune attacks from both parties.

IV. NOTATION AND PRELIMINARIES

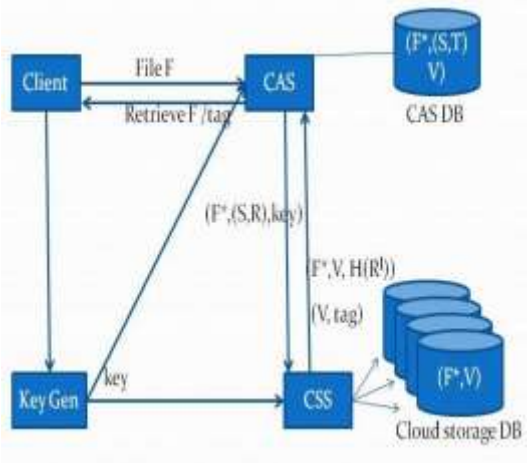
Merkle Hash Tree (MHT). The efficient and secure verification of the large data can be processed by Hash tree. It is used to check whether the block of data is unaltered and damaged or not. The Merkle hash tree construction is like as binary tree where secret values are associated to each leaf and non leaf nodes have hash values of its child nodes. Computing the Tree and Root hash as following steps-

- Select random secret 'B'.
- Derive leaf secrets Bi=h(B||i).
- Use hash ( ) function to get leaf/interior node value
- Publish root hash as R as the public key

The root value R is known by everyone as a public key. Behind of this notion is to partitioning the file into number of blocks. Apply hash to each block and combine iteratively until we get a tree with single Root Hash. Elliptic Curve Over Fp. Let p>3 must an odd number of prime. An Elliptic curve E in excess of Fp is defined in equation as follows y2=x3+ax+b Where a Fp, b Fp and 4a3+27b20 (mod p). The all points (x, y),xFp, yFp, which satisfies above equation. If P1 and P2 are on E, we can define A3 = A1 + A2.

V. SPOR SYSTEM MODEL

This architecture is similar to OPoR model with minor modifications. It has three major entities in the architecture, that are client, cloud audit server and cloud storage server. The key generator is an additional entity that generates a random temporary key for each request of client and sends it to both entities cloud audit server and cloud storage server. It can used to avoid the reset attacks of miscellaneous parties. The audit server and storage server check the key before to perform the operations on data for to avoiding individual miscellaneous operations. The file data encryption operation and decryption operation can perform by EC cryptography. The ECDH and ECDSA algorithms are used in process of key exchange and digital authentication, key generator algorithm used for random key generation. The EC cryptography groups can have a short length keys for encrypt, decrypt and its signature verification. It can provide speedup process using less memory and bandwidth savings with same profit of other cryptosystems. Let e: G1XG1→ GT is permissible bilinear pairing, generate Elliptic curve groups G1 and G2 of prime order p. Let cloud.



ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

Figure: 1 Structure of SPoR system process.

Audit and storage servers agree on (p, a, b, G, n, h) elements represent on elliptic curve. where p is a prime number and order of field Fp, a and b are coefficient constants of elliptic curve for defining equation, G is generator of large prime subgroup of point G(x, y) on the elliptic curve, n is order of G which is smallest positive number nG=, h is cofactor of Elliptic curve application EC (Fp)/n here h value range is less than or equal to four. The implementation and working of SPoR model can be categorized as four phases according to its process execution. Those are as following phases

A. Setup phase

In this phase the ECDH algorithm produce the key pairs as private keys Sk and Skl are random integers 1 to p-1 and generate public keys as Pk=SkG, Pkl=SklG for CAS and CSS.

B. Upload phase

The upload phase is divided into two phases that are client to CAS upload phase and CAS to CSS upload phase. Phase 1: Client to CAS. The client can upload the file F to the CAS, Key Generator generate a single and send to CAS and CSS. CAS encoded the file and divides into blocks B1, B2,...,Bs and generate a root R based on the Merkle hash tree construction, where the root leaf nodes are set of hashes of file blocks as H(Bi) where i=1,2,...,s. Next the CAS sign the root R with his private key Sk as h(R)Sk ← sig Pk(R). Hence the files tag t= sig Pk(R) is send to client as receipt. Phase 2: CAS to CSS

- The CAS computes the signature pairs(s, r) using ECDSA
- The CAS sends the processed file to CSS with its signature and key as (F*,(s, r),key).
- The CSS check the key of CAS with itself key for that file F* operation. CSS verify signature of file with s == v using (F*,(s, r),key). It generate root R using itself private key Skl and send v, H(Rl)as receipt to CAS of File F*. It store the file (F*,v) on database.
- The CAS checks the receipt value (v, H(Rl)) with its (s, H(Rl)) and Delete the local replica of File. It again send H(Rl) receipt to user as acknowledgment of File uploaded.

C. Integrity verification

In this process the client or CAS verify the outsourced data integrity by challenging the CSS. Select the file and send verification request.

- CSS on receiving request load the file (F*,v), calculate the signature of file and check v==s then send proof as(F*,v, H(Rl)) to cloud audit server.
- Upon receiving response from CSS, the CAS checks the vl==v then
- Check the consistency with generating signature of file and
- Check if s==v then file not modified. Otherwise it is modified.

D. Update phase

- Client sends update request to CAS, key generator send key to both CAS and CSS.
- The CAS generate corresponding signature based on its update file F* request and send (F*,(s, r), key) to CSS.
- The CSS check the key with itself key for that file operation then verify the file signature with s==v and replace the old file with new one, generate root R for updated file and send the receipt to CAS as (v, H(Rl)).
- The CAS check its receipt v==s then delete the local replica and send H(Rl) to client as acknowledgement of update operation.

VI. PERFORMANCE ANALYSIS

The SPoR system can developed by cloudSim tool, which can provide a cloud computing environment with cloud storage and audit server. The main advantage of Elliptic curve is to use small length keys and provide same standard of security as RSA and Diffie Hellman key pairs. In hardware binary curves are really fast so that it can have very fast key generation and moderately fast decryption and encryption. At integrity verification to speed up the process by storing vl with file which is directly compared by v, which is stored on audit server. The audit server deletes local replica of files after the verification of file tags.

The reset attacks is resisted by the unique key which is send to audit and storage server for upload and update file operations. Hence instead of key the storage server not commit the operations on file F*. The verification response time can evaluate with the parameters of file searching time, root generation time and signature generation time of CAS and CSS process.

VII. CONCLUSIONS

This SPoR system resist the reset attacks of CAS and CSS by generating unique key for each operations performed by them as per user request. Verification process speed is increased by Elliptic curve cryptography and it also reduces the cost of bandwidth and tag size. It verifies the tag with file before deleting the local replica of file. It also supports dynamic operations of data and public verifiability. The SPoR model is developed by using Merkle Hash Tree scheme for dynamic operations of integrity verification process on cloud. The MHT is not work on multi-cloud environment. Hence this paper leaves the development of this issue as future work.

REFERENCES

[1] A.Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 584–597.

[2] H.Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 14th Int. Conf. Theory Appl. Cryptol. Inf. Security, 2008, pp. 90– 107.

[3] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in Proc. ACM Workshop Cloud Comput. Security, 2009, pp. 43–54.

[4] H. Li, B. Wang, and B. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43–56, Jan.–Mar. 2014.

[5] J. Li, X. Tan, X. Chen, and D. S. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in Proc.5th Int. Conf. Intell. Netw. Collaborative Syst., 2013, pp. 93–98.

[6] C.Wang, Q. Wang, and K. Ren, "Ensuring data storage security in cloud computing," in Proc. 17th Int. Workshop Quality Serv., 2009, pp. 1–9.

[7] Q.Zheng, and S. Xu, "Secure and efficient proof of storage with deduplication," in Proc. ACM Conf. Data Appl.Security Privacy, 2012, pp. 1–12.

[8] C.Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. (2008). "Dynamic provable data possession", Cryptology ePrint Archive, Report 2008/432 [Online]. Available: http://eprint.iacr.org

[9] S. Maria Celestin Vigila, K. Muneeswaran "Elliptic curve based key generation for symmetric encryption," in ICSCCN international IEEE conference. year-2011, pp.824-829.

[10] The Elliptic Curve Digital Signature Algorithm (ECDSA). Don Johnson and Alfred Menezes and Scott Vanstone. Certicom Research, Canada. Dept. of Combinatorics & Optimization, University of Waterloo, Canada

[11] K.Ramesh, S.Ramesh, " Implementing OneTimePassword based security mechanism for securing personal health records in cloud," in ICCICCT international IEEE conference July 2014, pp.968-972.

[12] Yuan-Bin Xie; Pei-Jun Ma; Jiang-Yi Shi; Kang Li; Xiao-Feng Yang; Yue Hao "High-speed and flexible elliptic curve cryptographic processor for general prime fields", in ICSICT IEEE 10th international conference, year 2010, pp.503-505.

[13] The OpenSSL Project, see http://www.openssl.org/.

[14] Mahesh S.Giri " A Survey on Data Integrity Techniques in Cloud Computing " International Journal of Computer Applications (0975 – 8887) Volume 122 – No.2, July 2015

[15] Hero Modares, Amirhossein Moravejosharieh, Rosli Salleh "Wireless Network Security Using Elliptic Curve Cryptography", Informatics and Computational Intelligence (ICI), 2011 First International Conference

on Year:2011 Pages: 348 - 351, DOI: 10.1109/ICI. 2011.63

[16] S. V. Divya, R. S. Shaji "Security in data forwarding through elliptic curve cryptography in cloud "Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on Year:2014 Pages: 1083 - 1088, DOI: 10.1109/ ICCICCT. 2014.6993122

[17] Haichun Zhao, Xuefeng Zheng "A Survey on the Integrity Checking of Outsourced Data in Cloud Computing" 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-Scal Com) Year: 2015 Pages: 1650 - 1656, DOI: 10.1109/ UIC-ATC-ScalCom-CBDCom-IoP.2015.300

[18] Trushna S Khatri, G B Jethava "Improving dynamic data integrity verification in cloud computing" Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on Year: 2013 Pages: 1 - 6, DOI: 10.1109/ ICCCNT.2013.6726483